Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/US04/017438

International filing date: 01 July 2004 (01.07.2004)

Document type: Certified copy of priority document

Document details: Country/Office: US
Number: 60/483,629
Filing date: 01 July 2003 (01.07.2003)

Date of receipt at the International Bureau: 16 August 2004 (16.08.2004)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)

1210202

THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

August 11, 2004

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE.

APPLICATION NUMBER: *60/483,629*
FILING DATE: *July 01, 2003*
RELATED PCT APPLICATION NUMBER: *PCT/US04/17438*

Certified by

Jon W Dudas

Acting Under Secretary of Commerce
for Intellectual Property
and Acting Director of the U.S.
Patent and Trademark Office

Please type a plus sign (+) inside this box → +

PTO/SB/16 (8-00)
Approved for use through10/31/2002. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

60/483629 07/01/03

INVENTOR(S)

Given Name (first and middle [if any])	Family Name or Surname	Residence (City and either State or Foreign Country)
William M.	SHVODIAN	1301 Capulet Court McLean, VA 22102

☒ Additional inventors are being named on the 1 separately numbered sheets attached hereto

TITLE OF THE INVENTION (280 characters max)

Synchronization of Isochronous Streams Over a Wireless Communications Link

Direct all correspondence to:

CORRESPONDENCE ADDRESS

☒ Customer Number 29101 → Place Customer Number Bar Code Label here 29101 PATENT & TRADEMARK OFFICE

Type Customer Number here

OR

☐ Firm or Individual Name

Address					
Address					
City		State		ZIP	
Country		Telephone		Fax	

ENCLOSED APPLICATION PARTS *(check all that apply)*

☒ Specification Number of Pages 20

☐ Drawing(s) Number of Sheets

☐ Application Data Sheet. See 37 CFR 1.76

☐ CD(s), Number

☐ Other (specify)

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT *(check one)*

☐ A check or money order is enclosed to cover the filing fees

☒ The Commissioner is hereby authorized to charge filing fees or credit any overpayment to Deposit Account Number: 50-1869

☐ Payment by credit card. Form PTO-2038 is attached.

FILING FEE AMOUNT ($) $160.00

The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.

☐ No.

☒ Yes, the name of the U.S. Government agency and the Government contract number are: NIST ATP 70NANB2H3001

Respectfully submitted,

SIGNATURE Brian C. Altmiller Date 7/1/03

TYPED or PRINTED NAME Brian C. Altmiller

REGISTRATION NO. (if appropriate) 37,271

TELEPHONE (703) 269-3063

Docket Number: XSI.089P

USE ONLY FOR FILING A PROVISIONAL APPLICATION FOR PATENT

PROVISIONAL APPLICATION COVER SHEET
Additional Page

PTO/SB/16 (8-00)
Approved for use through 10/31/2002. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Docket Number	XSI.089P	Type a plus sign (+) inside this box ⟶	+

INVENTOR(S)/APPLICANT(S)		
Given Name (first and middle [if any])	Family or Surname	Residence (City and either State or Foreign Country)
Joel Z.	APISDORF	12205 Nutmeg Lane Reston, VA 20191

Number 1 of 1

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Synchronization of Isochronous Streams Over a Wireless Communication Link

Inventors: William M. Shvodian and Joel Z. Apisdorf

Docket No.: XSI.089P

Synchronization of Isochronous Str ams Over a Wir less Communication Link

- This invention applies timestamps to time sensitive packets arriving via a wired interface. The packets are queued and transmitted across the air including the timestamps. On the receiving side, the packets are fed out to the wired medium based on the timestamps. The timestamps are relative to a synchronizing event, such as the beacon in an 802.15.3 type of wireless network.
- Sending packets over the air changes their timing. However, in some applications, timing of packets is critical. Rather than struggle to maintain the proper timing over the air, this invention makes use of timestamps to preserve the timing information. The packets are transmitted over the air, then the time relationship is recreated on the receiving side.
- A preferred embodiment of this invention uses the 802.15.3 beacon as global time reference to allow devices to be synchronized with each other.
- In alternate embodiments, the beacon could be replaced by an alternative synchronizing event.
- Without the use of over-the-air timestamps, the timing relationship of packets arriving at the transmitting radio cannot be maintained at the output of the receiving radio.

TM 146.4 MAC Stream Synchronization

Title	TM146.4 MAC Stream Synchronization
Date	
Source	Bill Shvodian
Rev	
Abstract	
Summary	This tech memo describes Four methods for synchronizing streams so that timing on the transmitting DEV is maintained at the host interface of the receiving DEV.

Overview

- Background
- Option 1) beacon number/offset timestamp
- Option 2) Free running clock timestamp
- Option 3) Free running timestamps converted to beacon number/offset timestamps before transmission, and/or beacon number/offset timestamps converted to free running timestamps in the receiver
- Option 4) Free running timestamps used with a single beacon number/timestamp appended.
- Conclusions

Background

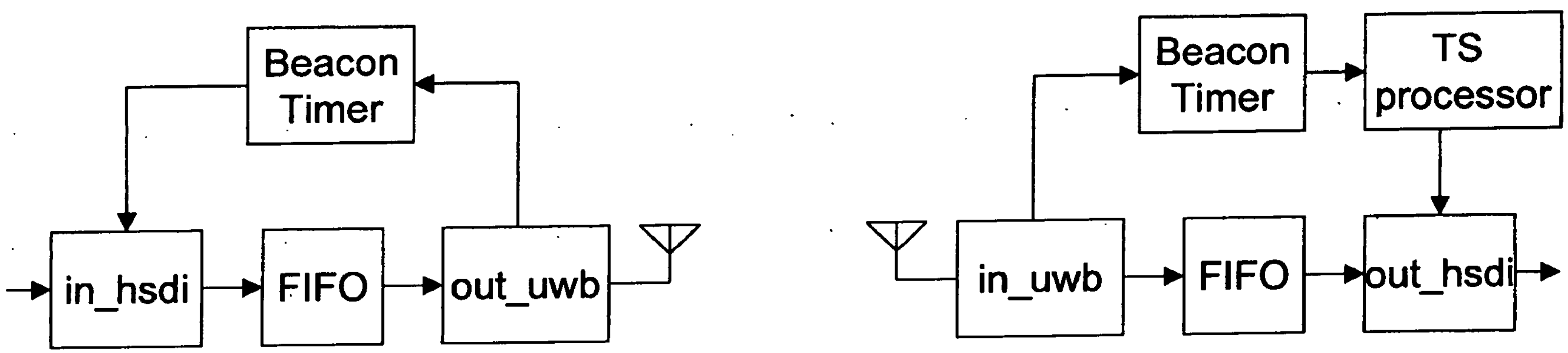
- Data coming in over an interface like 1394 needs to preserve the timing relationship on the incoming stream and reproduce it on the outgoing stream.
- Since the time relationship cannot be maintained over the wireless link itself, some sort of timestamping of packets is required.

  Relative timestamps (arrival delta between packets) doesn't work because the receiver clock and the transmitter clock are not synchronized. This will cause the receiver buffer to overflow or underflow depending on the relative clock rate.
- Three methods of absolute time stamping will be described here. One uses the beacon number and offset from the start of the beacon. The other uses a free running timer at the sending and receiving DEV. The third is a hybrid between the two.

Option 1) Beacon Number/Offset timestamps

- Incoming cells or packets are timestamped upon arrival on hsdi
- Timestamps consist of a 802.15.3 beacon number (16 bit) and offset from the start of the superframe in μs (16 bit).

  The receiving DEV adds a time offset to the timestamp and feeds out the cells to the hsdi based on that timing



Option 1) HW/SW processing required

- in_hsdi HW needs to know the beacon number and offset into the superframe to put into the timestamp.

  Out_hsdi HW needs to know the beacon number and offset into the superframe to feed packets onto the hsdi.

  HW knows when superframe should have started. Beacon number is incremented and offset is reset to zero at the end of the superframef
- When Beacon HCS passes at the receiver, offset timer is reset to $T_{offset}=T_{PA}+T_{Hdr}+T_{latency}$
- In_uwb and out_uwb needs to be modified to process beacon number/offset timing

  Beacon number needs to be initialized with value from a beacon

Option 1) 15.3 timestamp Frame Format

HSDI_len	HSDI_sig	HSDI_TS	HSDI_1

hsdi_pkt

PHY_HDR	MAC_HDR	HCS	FCSL_HDR	HSDI_pkt_1	HSDI_pkt_2	HSDI_pkt_3	FCS

15.3 frame

Option 1) Time Stamp Generation

There are a few ways to generate the Beacon number/offset timestamp

A. Offset continues to increment until the next synchronizing event
   - Extra bits required for the offset field
   - Even with extra bits, missing a number of synchronizing events eventually causes the offset field to wrap
   - Inefficient use of bits because the offset overlaps with the beacon number field even with max size superframe

B. Offset wraps to zero after the end of the superframe
   - The number of offset bits required to cover the max superframe duration is all that is needed. No extra bits are required for the offset field
   - Beacon number is either incremented by HW or loaded with a value set by SW.

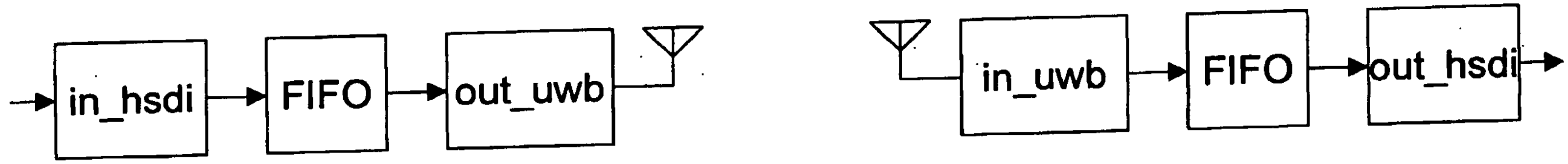
Recommend Option 2.

Option 2) Free Running Timestamp

- Incoming cells or packets are timestamped upon arrival on hsdi
- Timestamps consist of a 32 bit free running .1 μs timer (could be 1 μs or 1/16 μs)
- Beacons are also time-stamped upon arrival

  Beacon timestamps are encapsulated along with the packets in each frame.
- The receiver process the beacon timestamps and correlates with its own time base.

  The receiver calculates the time to release the cell



Option 2) Free-running timestamp Frame Format

HSDI_len	HSDI_sig	HSDI_TS	HSDI_1

hsdi_pkt

PHY_HDR	MAC_HDR	HCS	FCSL_HDR	Bcn_TSs	HSDI_pkt_1	HSDI_pkt_2	FCS

15.3 frame

Option 2) HW/SW processing required

Tx:

- in_uwb writes absolute free-running 32 bit TS of SFD for every packet.
- SW copies beacon TS to beacon TS array
- SW knows when beacon should have arrived
- SW will copy frame TS to beacon array if frame is a beacon
- SW will copy beacon TS array to out_uwb ahead of uwb frame (fixed or variable size field?)

Rx:

- SW will copy Tx beacon TS array to HW memory
- SW will copy Rx beacon timestamps to HW memory
- Out_HSDI will read TS of next frame
- SW will fill the beacon FIFO with all relevant TS for each DMA
- 4 variables T_{SA} = Beacon time at Tx, T_{SB} = beacon time at Rx, T_C = cell time T_L = latency. T_R is the time the cell should be released on HSDI
- $T_R = T_C - T_{SA} + T_{SB} + T_L$
- For each pkt, the out_hsdi needs to check the timestamp against the timestamp of the next beacon.

Option 2) Free running TS Issues

Rx needs to be able to correlate TX beacon timestamps to Rx beacon timestamps. May need to add beacon number to the TS so there is no discrepancy.

- Receiving 2 separate streams may require a separate DMA

What happens if no beacons received since the last frame? Is not beacon TS array sent?

- How do we ensure that beacon array is sent with multiple streams?

Does Rx need to store beacon TS array for each stream?

What resolution TS should we use? 1 µs or .1 µs?

- TS array will be variable size. Could transmit a maximum number of beacon TSs.

General discomfort about maintaining the beacon timestamp array and sending it in the frames.

Comparison

15.3 TS requires HW mods only. Free running TS requires both HW and SW mods.

Free running TS requires SW operation on every frame. For each pkt a comparison needs to be made between the HSDI timestamp and the next Tx beacon timestamp.

Free running TS will have more overhead
- Beacon TS array is also transmitted.
- Free running TS method may also require sending a beacon number and TS in the beacon TS array so that Rx beacon TSs can be correlated to TX beacon TSs. Maybe only one beacon number sent per frame.

Option 3) Free running TS converted to 15.3 timestamps

- Tx can use either 802.15.3 TS or free running TS at the Tx DEV or at the Rx DEV. Choice is up to the implementer
- If free running TS are used, the transmitting DEV converts to 15.3 frame number/offset time before transmission.
- Frames sent across the air have the same format as Option 1).
- The receiving DEV has the option of using 15.3 timestamps or converting to free running timestamps for packets going out to the HSDI

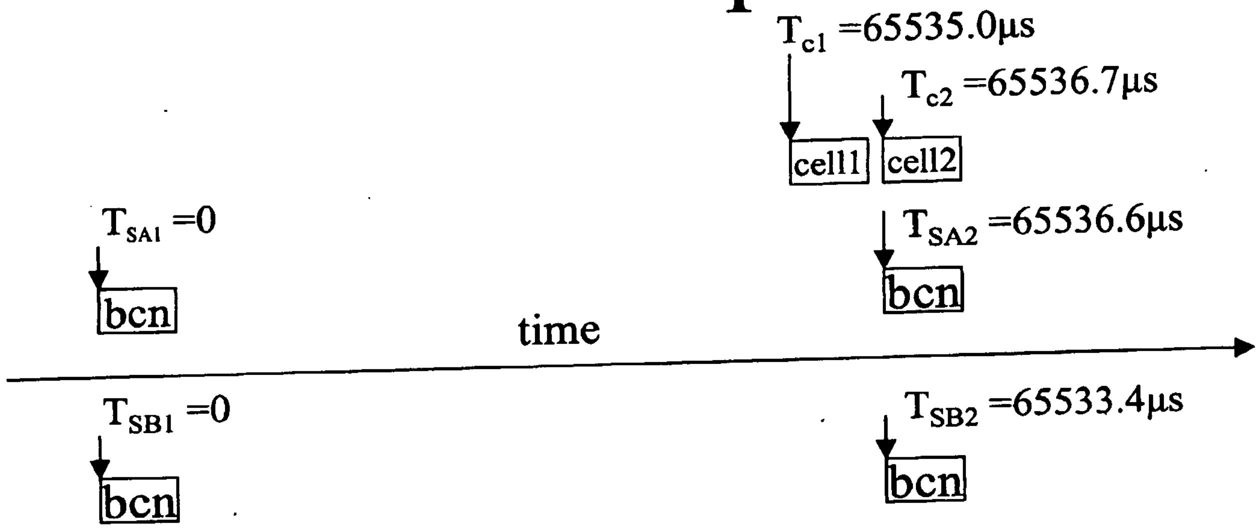
Non-monotonic TS at out_hsdi Option 1

If a cell arrives on the HSDI just before the synchronization event, the cell just after the synchronization event may have a time stamp that earlier in time than the timestamp of the cell that arrived just prior to the synchronizing event

Out_hsdi must realize that the time is in recent past and not the distant future and send the cell over the HSDI immediately

The same issue occurs in Option 2

Non-monotonic TS at out_hsdi Option 2



At output HSDI:

- $T_{R1} = T_{c1} - T_{SA1} + T_{SB1} + T_L = 65535.0 + T_L$
- $T_{R2} = T_{c2} - T_{SA2} + T_{SB2} + T_L = 65536.7 - 65536.6 + 65533.4 + T_L = 65533.5 + T_L$

- T_{R2} is less than T_{R1} so option 2 does not prevent non-monotonically increasing release times

Timestamp Details

- 32 bit timestamp
  - 12 bit superframe number plus 20 bit offset (16 bits for superframe µs offset, plus 4 bits for 1/16 µs granularity)
- Timestamp is appended to each packet

Option 4) Free running timestamp

- 32 bit free running timestamp attached to each frame.
- Timestamp is 16 MHz clock, 1/16 us per tick
- Each frame also contains just one 16 bit beacon number and a 32 bit timestamp for when that beacon was received.
- Receiver uses the beacon number timestamp to convert the time reference from the sender's time reference to the receiver's time reference
  – Software calculates the value for the packet to be released FRT(TxPkt) + (latency + (FRT(RxSE)-FRT(TxSE)

Conclusion

Option 4 chosen to simplify HW and put as much into SW as possible for flexibility.